AMENDMENTS TO THE CLAIMS

Re-write the claims as set forth below. This listing of claims will replace all prior versions and listings, of claims in the application:

1. (Currently Amended) A method for static single assignment form dead code elimination, the method comprising:

examining a first instruction of a machine code off of a worklist in memory, wherein the first instruction includes a previous link and a write mask;

examining at least one second instruction, wherein the at least one second instruction is a source of the first instruction and wherein each of the at least one second instruction includes a previous link and a write mask;

determining, using said previous link of said at least one second instruction, if any components within a particular field of the at least one second instruction are required, wherein said previous link of said second instruction links said second instruction with a prior instruction that writes at least one component of said components;

when no components of the at least one second instruction are required, deleting the at least one second instruction from the machine code; and

when any component of the at least one second instruction is required, adding the at least one second instruction to the worklist in the memory.

2. (original) The method of claim 1 further comprising:

generating the worklist by:

for each of a plurality of instructions, determining if the instruction is a critical instruction; and

if the instruction is a critical instruction, writing the instruction to the worklist.

3. (original) The method of claim 2 further comprising:

setting a live bit for each component of the plurality of instructions.

4. (previously presented) The method of claim 2 wherein each critical instruction is an instruction that generates an export value.

5. (previously presented) The method of claim 2 further comprising:

prior to generating the worklist:

receiving a plurality of instructions;

adding to each instruction the previous link; and

adding to each instruction the write mask.

6. (original) The method of claim 5 wherein the write mask is a multi-bit field representing a number of components in a superword register.

7. (previously presented) The method of claim 6 wherein each of the plurality of instructions are written to the worklist a predetermined number of times, wherein the predetermined number of times is based on the number of components in the superword register.

8. (Currently Amended) A method for static single assignment form dead code elimination comprising:

receiving a plurality of instructions;

adding to each instruction a previous link, wherein said previous link links said each instruction with a prior instruction that writes at least one component;

adding to each instruction a write mask; and

generating a worklist in memory by:

for each of the plurality of instructions, determining if the instruction is a critical instruction; and

if the instruction is a critical instruction, writing the instructions to the worklist in the memory.

9. (previously presented) The method of claim 8 further comprising:

setting a live bit for each component of the plurality of instructions;

examining a first instruction off of the worklist;

examining at least one second instruction in the machine code, wherein the at least one second instruction is a source of the first instruction;

determining if any component within a particular field of the at least one second instruction is live; and

when no components of the at least one second instruction are live, deleting the second instruction from the worklist.

10. (cancelled)

11. (previously presented) The method of claim 8 wherein each critical instruction is an instruction that generates an export value.

12. (original) The method of claim 8 wherein the write mask is a multi-bit field representing a number of components in a superword register.

13. (previously presented) The method of claim 12 wherein each of the plurality of instructions are written to the worklist a predetermined number of times, wherein the predetermined number of times is based on the number of components in the superword register.

14. (Currently Amended) An apparatus for static single assignment form dead code elimination comprising:

at least one memory device storing a plurality of executable instructions of a machine code; and

at least one processor operably coupled to the at least one memory device, operative to receive the plurality of executable instructions such that the at least one processor, in response to the plurality of executable instructions is further operative to:

examine a first instruction off of a worklist, wherein the first instruction includes previous link and a write mask;

examine at least one second instruction of the machine code, wherein the at least one second instruction is a source of the first instruction and each of the at least one second instruction includes a previous link and a write mask;

determine, using said previous link of said at least one second instruction, if any component within a particular field of the at least one second instruction is live, wherein

said previous link of said second instruction links said second instruction with a prior instruction that writes at least one component of said components; and

when no components are live, delete the second instruction from the machine code.

15. (previously presented) The apparatus of claim 14 wherein the at least one processor in response to the plurality of instructions executable instructions is further operative to :

generate the worklist by:

for each of a plurality of instructions, determining if the instruction is a critical instruction; and

if the instruction is a critical instruction, writing the instruction to the worklist.

16. (previously presented) The apparatus of claim 15 wherein each critical instruction is an instruction that generates an export value.

17. (cancelled)

18. (previously presented) The apparatus of claim 15 further comprising:

a superword register operably coupled to the at least one processor, wherein the write mask is a multi-bit field representing a number of components in the superword register.

19. (Currently Amended) An apparatus for static single assignment form dead code eliminations comprising:

at least one memory device storing a plurality of executable instructions of a machine code; and

at least one processor operably coupled to the at least one memory device, operative to receive the plurality of executable instructions such that the at least one processor, in response to the executable instructions is further operative to:

receive a plurality of instructions;

add to each instruction a previous link, wherein said previous link links said each instruction with a prior instruction that writes at least one component;

add to each instruction a write mask; and

generate a worklist by:

for each of the plurality of instructions; determining if the instruction is a critical instruction; and

if the instruction is a critical instruction, writing the instructions to the worklist;

examine a first instruction off of the worklist;

examine at least one second instruction from the machine code, wherein the at least one second instruction is a source of the first instruction;

determine, using a previous link of said at least one second instruction, if any component within a particular field of the at least one second instruction is live; and

when no component is live, delete the second instruction from the machine code.

20. (original) The apparatus of claim 19 further comprising:

a superword register operably coupled to the at least one processor, wherein the write mask is a multi-bit field representing a number of components in a superword register.

21. (previously presented) The apparatus of claim 15, wherein the at least one processor in response to the plurality of executable instructions is further operative to set a live bit for each component of the plurality of instructions.

22. (previously presented) The method of claim 9, further comprising:

when any component of the at least one second instruction is live, adding the at least one second instruction to the worklist.

23. (previously presented) The apparatus of claim 14, wherein the at least one processor in response to the plurality of instructions executable instructions is further operative to :

when any component of the at least one second instruction is live, add the at least one second instruction to the worklist.

24. (previously presented) The apparatus of claim 19, wherein the at least one processor in response to the plurality of instructions executable instructions is further operative to :

when any component of the at least one second instruction is live, add the at least one second instruction to the worklist.